Combining Structured and Unstructured Data in Electronic Health Record for Readmission Prediction via Deep Learning

Shijie Qu

THE OHIO STATE UNIVERSITY

Email: qu.219@osu.edu

Phone: (614)316-8827

Supervisor: Prof. Ping Zhang

Presented in Partial Fulfillment of the Requirements for the Degree B.S. in Computer Science and Engineering with Honors Research Distinction in the College of Engineering at Ohio State University

© Copyright by
Shijie Qu
2020

Acknowledgements

I would like to give immense thanks to Professor Ping Zhang for accepting me into his research group as the only undergraduate student. I greatly appreciate the help and support Professor Zhang provides throughout the entire period of the research. Also, I would like to express my gratitude to Dr. Ping Zhang, Dr. Rajiv Ramnath, and Dr. Theodore Allen for serving on my thesis committee.

Vita

Dec 6th, 1997 . . . . . Born in Beijing, China

Sep 2013 - Jun 2016 . . . . . Affiliated High School of Peking University

Aug 2016 - May 2020 . . . . . B.S. CSE, Ohio State University

Fields of Study

Major Field: Computer Science and Engineering

Second Major Field: Data Analytics

Minor Field: Economics

Abstract

With the aid of statistical learning tools nowadays, a variety of clinical prediction tasks can be examined and modeled in a quantitative way. Predicting hospital readmission probability is among one of the most significant tasks in that it provides a good indication of the healthcare cost and a patient's health condition. Thus, in this study, we strive to build up a quantitative prediction model for readmission prediction by utilizing both structured data and unstructured text data from a patient's Electronic Health Records (EHR). In the past, a variety of studies focused on using only structured categorical or numerical data such as lab tests and Heart Failure Signs to perform clinical risk prediction tasks, while recently, with the help of deep learning models, people started to use Natural Language Processing techniques to process unstructured patient's text data, as it contains richer information. However, with the belief that both structured and unstructured data can play significant role in predicting readmission, our research will focus on developing deep learning methods to combine the two types of data together in an efficient way, such that the predicting performance will exceed those of the previous models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic Health Record (EHR) provides a systematic and structured way to store a patient's information. In a typical EHR sample, various type of data is included, such as a patient's demographic data (e.g., age, gender, etc.), medical codes (e.g., Standard Disease Codes, National Drug Codes, etc.), and lab test results (e.g., blood pressure, heart rate, etc.). An example of a typical EHR is shown below in Figure 1.1. Typically, an EHR record includes a wide range of knowledge related to a patient's status before, during, and after he or she is admitted into the hospital. Due to the comprehensive information stored in EHR, it can be used on several occasions to help improve healthcare services. Some of the exemplar clinical prediction tasks which researcher conduct using EHR data include the prediction of length of stay in the hospital, in-hospital mortality, and readmission of patients. [1][2]

To perform the clinical tasks based on EHR data, traditionally, people would create a couple of rule-based systems and manually-selected features. However, with the recent burgeon in deep learning, more and more research started to leverage neural-based frameworks to generate better representations of the input data, and use the intermediate representation to yield better regression or classification performance. [3][4] However, due to the sparsity and multi-modality nature of EHR data, representational learning

in the past days were not able to achieve satisfactory results in clinical prediction tasks as it did in other disciplines such as image and audio processing.

One of the limitations of previous work is that only a small fraction of the entire EHR record is utilized, where the data is more cohesively related and are stored in a similar format. For instance, performing pneumonia risk prediction using patient's lab test results [5], or predicting chronic disease using clinical notes [6]. To address this issue, our research strives to combine different modalities of data in EHR and come up with a more reliable model for clinical risk prediction. In this research, the primary clinical prediction task we focus on is 30-day readmission probability. The definition of 30-day readmission will be provided in the later section.

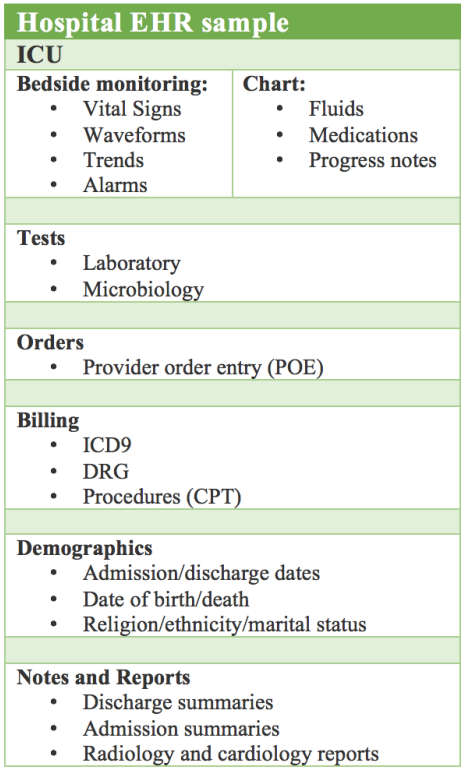

Figure 1.1: EHR Example

1.1 Contributions and Significance

As described above, this research aims at discovering efficient methods to employ both structured data and unstructured data in predicting hospital

30-day readmission probability. To accomplish this goal, we present the following two distinct approaches:

ICD-based text representation: Our first approach is built upon an attention-based neural model proposed by *Mullenbach et al* [7]. The original model used self-attention mechanism to perform ICD-code [1] classification by feeding in clinical note as input. To adjust for our task in readmission prediction, we transfer part of the model, which takes text as input and generate intermediate text representation by letting the input go through the ICD-code attention layer. In this way, we incorporate the knowledge of ICD-code into the text. We will demonstrate that this approach outperforms those that use only text or ICD-codes for prediction.

Attention-based knowledge linking: This approach allows us to "connect" structured information, including lab results and vital signs, to unstructured discharge notes. The model consists of a transformer encoder to encode texts, an LSTM unit to encode structured input, and another LSTM unit to process the mixed data. The structured and unstructured information are being linked together by attention mechanism. It turns out that this linking network achieves a much better result than most of the state-of-the-art methods do.

1.2 Pertinent Concepts

Throughout this paper, the following terminologies related to biomedical informatics, deep learning or statistics will be repeatedly mentioned:

[1] International Statistical Classification of Diseases and Related Health Problems (ICD), a medical classification code created by the World Health Organization.

30-day Hospital Readmission: 30-day readmission refers to the unplanned admission of a patient to an acute care hospital in 30 days after he/she is discharged from a hospital last time. The readmission rate could, to some extent, serve as a measurement for a hospital's healthcare quality, and an accurate prediction of the readmission probability could prevent a hospital from being penalized.

Convolutional Neural Network (CNN): A type of Artificial Neural Network(ANN) designed initially for image processing. It typically consists of a combination of convolutional layers, pooling layers, fully-connected layers, and normalization layers.

The convolutional layer is literally the core of CNN, as it helps filter out the local information that is considered significant. The filter (or kernel) is normally represented by a $K \times K$ matrix, and the way we perform convolution is to slide the kernel matrix horizontally and vertically on our data matrix. We compute element-wise multiplication and add the output together. The final output is considered the filtered output of the local $K \times K$ area. Then, we apply a non-linearity function to the filtered data, due to the fact that most data are non-linear. Pooling Layer comes next, to reduce down the dimensionality while preserving the most important information. In practice, the convolution-nonlinearity-pooling sequence is repeated more than one time to process the data thoroughly before passed everything into a linear classification layer. The basic workflow of CNN is shown in the figure below.

The characteristic of CNN enables it to focus on local or nearby information that is coherently related to each other. With its growing popularity

nowadays, CNN is being adopted in various domains such as Language Processing and Recommender Systems.

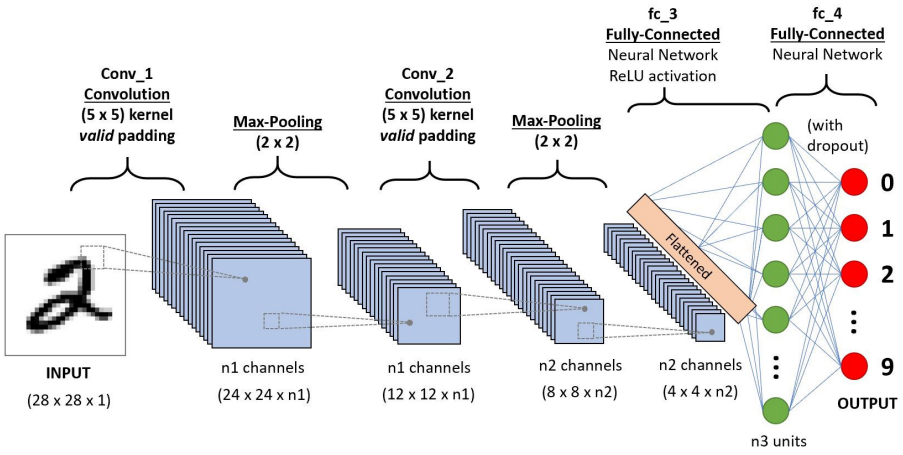

Figure 1.2: Convolutional Neural Network

Long Short-Term Memory Network(LSTM): LSTM is another type of artificial networks, but is specifically beneficial when processing temporal data. To handle the vanishing gradient problem in vanilla Recurrent Neural Network, LSTM refines the network by adding three "gates" to each LSTM cell: an input gate, an output gate, and a forget gate. The gating mechanism is built upon matrix element-wise multiplication, and nonlinear functions to achieve the functionality of forgetting or remembering a certain amount of information from the past. A more detailed examination of the cell structure is shown in figure 1.3.

Each cell stands for a timestamp. Thus, to represent multiple timestamps, we will have several LSTM connected sequentially. As a result, the input to each cell includes both the data at current time and the output

from the previous cell. In other words, the current cell will learn both the knowledge at current timestamp, but also everything before it.

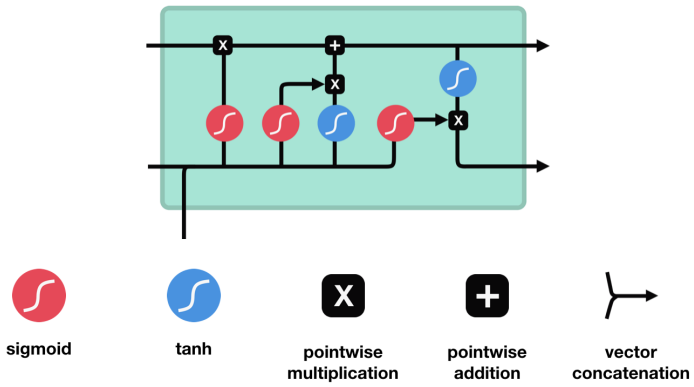

Figure 1.3: Long Short-Term Memory Network

Attention Mechanism: The idea of attention is brought up in the famous paper *Attention Is All You Need* [8]. The general idea of attention is, as it literally indicates, to focus on something that is more important.

Consider the sentence "The animal didn't cross the street because it was too tired" in the figure below. Here, the word "it" refers to the word "animal". However, if we use conventional word embedding like skip-gram [9], we will only be able to learn the embedding of a word by its nearby words in a sentence. Thus, since the word "animal" is posited relatively far away from the word "it", it is hard to learn this type of semantic connection. The invention of attention mechanism simply tackles this problem by multiplying the original word vector of each word to the original word vector of the word 'it', to calculate the similarity score. Then, we do a weighted sum of each similarity score and its corresponding word vector, to come up with a refined

word embedding for the word 'it'. In this way, we successfully capture the information that is supposed to be more critical.

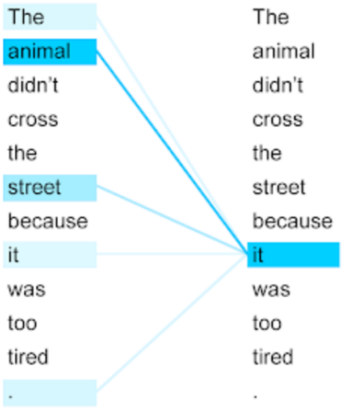

Figure 1.4: Attention Mechanism Example

1.3 Organization

This thesis is organized into five main chapters. The first chapter provides a gentle introduction of EHR mining and clinical risk prediction, as well as the contribution of our research. The second chapter begins with a review of current work in deep learning and clinical risk prediction tasks. Then, in chapter 3, the paper will examine the two approaches we developed to combine structure and unstructured data for readmission prediction. Chapter 4 will follow up with the results of the two approaches and discuss our findings. In the last chapter, we will give a brief conclusion of our work, and identify potential work to be done in the future.

Chapter 2

Related Work

Electronic Health Record (EHR) offers a comprehensive description of a patient's condition from him or her being admitted to the hospital until he or she is discharged, and the broad adoption of EHR in the recent years provides an unprecedented opportunity for researchers to conduct healthcare-related research.

Traditionally, most healthcare analysis utilized classical approaches for extracting features and designing rule-based systems. For instance, using keyword search to extract indicators for disease severity modeling [10], and implementing rule-based decision tree to predict asthma status of a patient [11]. Meanwhile, several knowledge-based scores are being introduced and adopted universally. SAPS III [12] was designed to measure the severity of disease for an ICU patient, and HOSPITAL Score [13] aimed at predicting avoidable readmission to the hospital.

With the rapid development of computing power and neural networks in the last few years, deep learning has demonstrated its effectiveness in a wide range of disciplines, such as speech [14] and video processing [15], image recognition [16] and machine translation [17].

Recently, healthcare research studies have also started to shift their focus to deep learning as well. As a result, various clinical predictions are being

done with the help of neural networks. *Mullenbach et el.* [7] proposed the idea of Self Attention Convolutional Neural Network to predict ICD codes by feeding in discharge notes. *Sushil et el.* [18], in their research, implemented a Stacked Denoising Autoencoder to perform mortality prediction as well as diagnosis classification. *Tang et el.* [19] opted to choose a recurrent network to capture the temporal structure of the lab results, in order to better predict patient's Length of Stay and Readmission probability. In a similar fashion, *Harutyunyan et el.* [20] came up with a multi-task LSTM model, which could handle various tasks simultaneously with satisfactory results. Overall, it is shown that deep learning models overwhelm traditional approaches in most clinical risk prediction tasks [21].

However, since it is generally cumbersome to integrate different data modalities together, very few studies have been done using both structured and unstructured data. Recently, a couple of research [6][22] have shown progress in using name-entity recognition(NER) to extract key entities from text, and combine them with the original structured data to perform the prediction. However, a potential drawback of this approach is that NER process requires specific domain knowledge to be supplied; Thus, the model needs to be adjusted every time we switch to a new task. Therefore, with the goal of efficiently combine structured and unstructured data without domain knowledge, we propose two distinct approaches in the next section.

Chapter 3

Methodology

3.1 Dataset Description

The data we used throughout the research comes from Medical Information Mart for Intensive Care III (MIMIC-III) dataset [23], a publicly available EHR dataset that contains data for over 40,000 de-identified ICU admissions at Beth Israel Deaconess Medical Center between 2001 and 2012. The dataset is comprised of different modalities of data, including but not limited to demographics, vital signs, laboratory results, medications, etc., which in total, takes up around 50GB of memory.

3.2 Code Implementation

The codes are implemented entirely in Python. Numpy and Pandas are the two libraries mainly used during the data preprocessing step. The former handles matrix processing while the latter one contributes to the manipulation of data frames. PyTorch library is the one we used to build up neural networks, as it provides the functionality of automatic differentiation and supports GPU computing.

3.3 Approach 1: ICD-based text representation

As introduced in the previous chapter, the ICD coding system serves as a worldwide standard for representing clinical procedures and diagnoses. By utilizing a hierarchical structure, ICD coding assigns each type of procedure or diagnosis a distinct code. For example, in the 9th edition of the ICD coding system, Asthma is represented by code 493.90, while Sepsis is denoted by the code 995.91. For billing purposes, these codes will be assigned to a patient by the hospital when the patient is ready to be discharged.

Meanwhile, along with the code, a patient will also receive a copy of his or her discharge summary, which records the patient's condition during the hospital admission, everything he or she went through in the hospital, as well as instructions after discharge.

Overall, ICD coding provides a tabular format of information that gives a precise outlook of a patient's status, while the discharge summary, on the other hand, could be unstructured and verbose, but may contain more subtle details of a patient. Therefore, we believe that a proper way to "mix" these two types of data together would yield a better risk prediction model for hospital readmission. The general idea of this approach is illustrated in Figure 3.1.

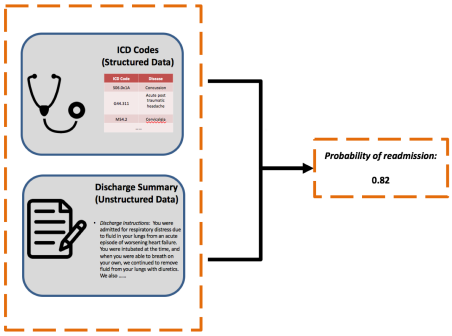

Figure 3.1: General Idea of Approach 1

3.3.1 Data Preprocessing

In total, there are about 60,000 patient samples in the MIMIC-III database. We first removed some extreme samples such as patients with age less than 18, and patients who die in the hospital. Next, we noticed that most of the patients do not have more than one admission; Thus, the number of positive samples is far less than the number of negative samples. To address this imbalanced sample issue, we first picked out all the 2114 positive patient samples. Then, for each positive sample, we match it to three negative patient samples that share similar age and the same gender with the positive sample. In this way, we successfully extract 8456 patient samples, with the positive-negative ratio of 1:3. Furthermore, to reduce dependency between admissions, we only kept the first admission record of a patient and remove all the others. For the ICD codes we used, we only kept the top-50 frequent ones, to avoid the data being too sparse. Regarding unstructured discharge notes, we followed the same preprocessing procedure introduced in this paper[7], where stopwords, infrequent tokens, and tokens with non-alphabetic characters are all removed. Documents are all being truncated or padded to the length of 2500 tokens. We also pretrained a word embedding using word2vec method [9], with embedding dimension of 128.

3.3.2 Model Architecture

The neural network is implemented based on the DR-CAML model[7], which appends a per-label attention mechanism to the Convolutional Neural Network(CNN). An illustration of the model is shown in figure 3.2 below.

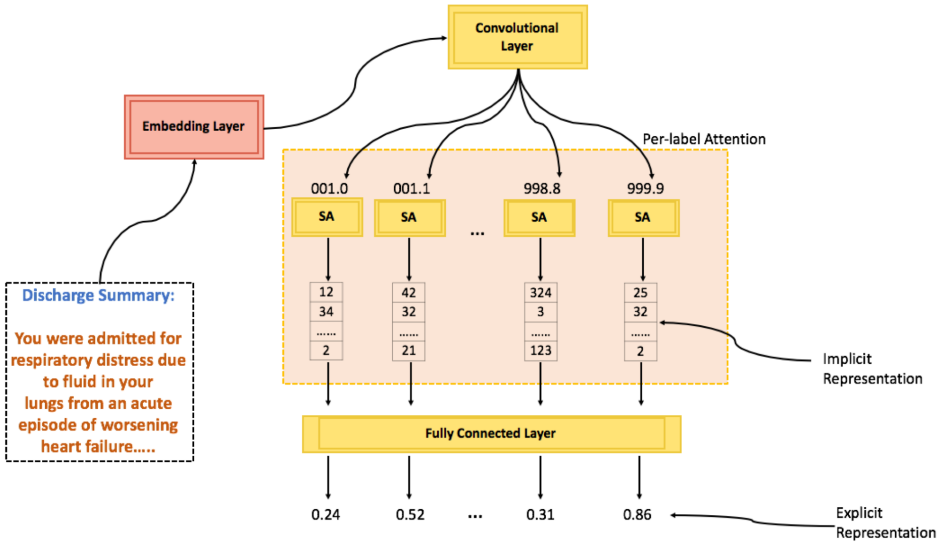

Figure 3.2: DR-CAML Model

The general idea of the model is to predict the probability of each ICD code by feeding in discharge notes as input. The input is represented by a series of word indices, each representing its corresponding word. Then, it is passed into an embedding layer, which uses the pretrained word2vec embedding to vectorize each word index, and thus generate a matrix $X \in R^{d_e \times N}$, where d_e is the embedding dimension. A convolutional layer is followed to capture the local features and create a new text representation $H \in R^{d_c \times N}$, where d_c is the filter output size. Next, we move on to the per-label attention layer, where for each label, the new text representation will be multiplied by the label's corresponding vector $u_l \in R^{d_c}$. As a result, after normalizing the output, we will have an attention score vector for each of the labels. Each element in the attention score vector stands for the "attention weight" assigned for the corresponding word (Attention mechanism is explained in

1.2). Lastly, for each ICD code, by multiplying H with its attention score vector, we will obtain a new representation for the text corresponding to that ICD code. This representation is named *implicit representation*, since it leverages attention mechanism to generate a new representation of the text that has absorbed the knowledge of each ICD code. For the original model, to get the probability for each ICD code, the implicit representation is fed into a fully connected layer, and then into a Softmax function for normalization. The probability of all 50 ICD codes generated by the model will be called *explicit representation*, as it directly uses probability to represent the knowledge learned.

Lastly, both implicit representation and explicit representation will be passed into a logistic regression function to predict the binary outcome of hospital readmission. We will compare the performance of implicit and explicit representation, to the performance of baseline methods in the next chapter.

3.3.3 Training Specifications

During the training process, the idea of 5-fold cross validation is adopted to split the training, validation and testing sets. In other words, we trained 5 different models, where each time 3 folds are used for training, 1 fold is used for validation, and the remaining one for testing. Cross validation generally help decrease the variance in the model performance.

Loss of the model is set to be the Binary Cross-Entropy loss, which is

defined in the following formula:

$$BCELoss = -\sum_{i=1}^{L}(ylog(p) + (1-y)log(1-p)$$

where p is the predicted probability, y is the true label, and L is the total number of samples.

Each word in the document is vectorized by a pretrained embedding using word2vec CBOW method [9] with embedding size 100. For hyperparameters of CNN, we chose to have 50 filter maps, with each filter size being 10. Dropout rate is set to be 0.2, and original learning rate is set to be 0.0001. Adam [24] is used for optimizing the learning rate. All the computations are done using GPU resource units from Ohio Supercomputer Center [25].

3.4 Approach 2: Attention-based knowledge linking

While the first approach concentrates more on utilizing ICD code to encode the text via attention mechanism, however, we noticed that in most cases, ICD codes are manually recorded by humans, which makes it subject to errors. In this case, we speculated that lab tests and vital signs should be more objective in depicting a patient's condition. Furthermore, instead of using attention to encode the text into a new representation, we would like to find a way to make the process more explicit, that is, to directly link the structured and unstructured data together. With all these thoughts in mind, we decided to build up another model to fulfill these goals.

The general idea is depicted in figure 3.3. The table on the top indicates the lab results or vital signs for a patient, for the first N hours of his or her admission, while the one below is an example of discharge note. The idea of approach 2 is to link each hour's structured data, with the most relevant words in the text, in order to perform the prediction task later.

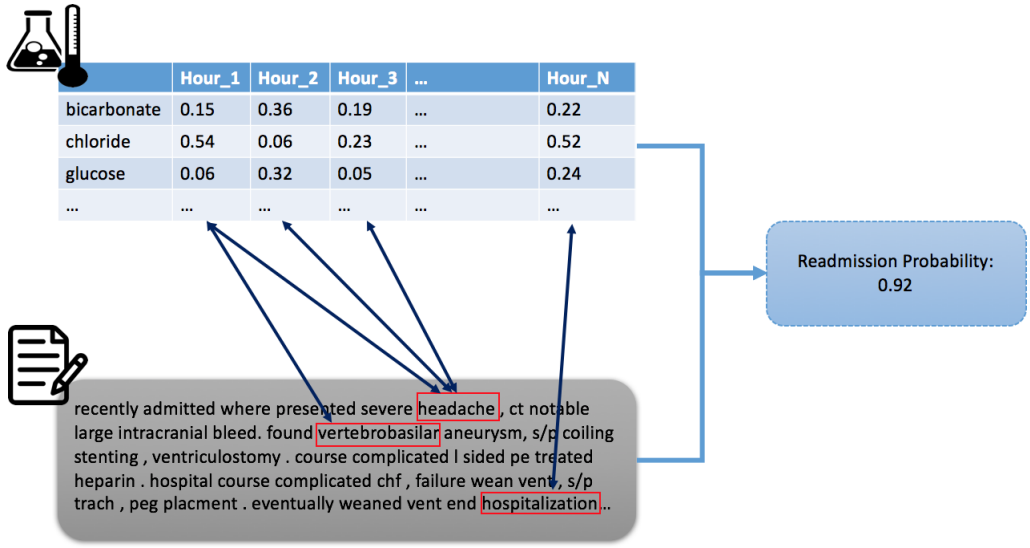

Figure 3.3: General Idea of Approach 2

3.4.1 Data Preprocessing

To make the input length to LSTM consistent, we only used the first 24 hours of a patient's vital signs and lab test results. The missing values are automatically replaced by 0.

We also consider adding the patient's demographic data such as gender, age, and ethnicity this time. We used one-hot encoding to encode the demographic data.

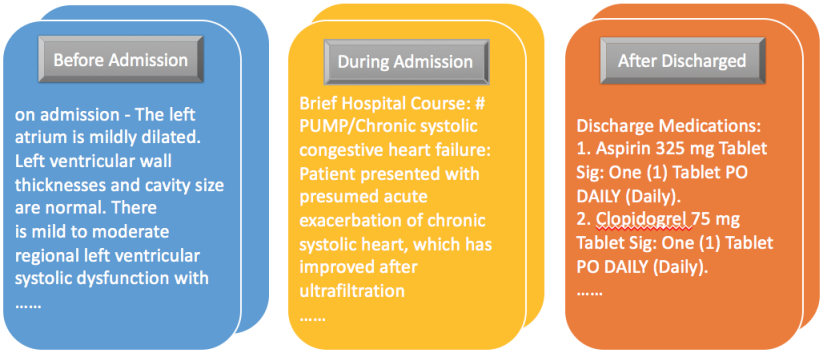

Figure 3.4: Text Preprocessing Example

For text processing, to make the prediction more accurate, we decided to go one step further than we did in section 3.3.1. We noticed that even though the discharge summary is considered unstructured data, it always follows some patterns in formatting. For example, in most cases, it starts with a patient's illness history, to a summary of the patient's activity in the hospital, then to some post-discharge instructions. Thus, we decided to split each discharge summary sample into three categories, "Before Admission", "During Admission", and "After Discharged". An example is shown in figure 3.4.

3.4.2 Model Architecture

Unstructured representation module

To process text data, we used the state-of-the-art Transformer Encoder model [8]. Encoder primarily leveraged the idea of multi-head self-attention mechanism to generate a better embedding of a word by allowing a word to interact with other words in a sentence to know which word to pay more

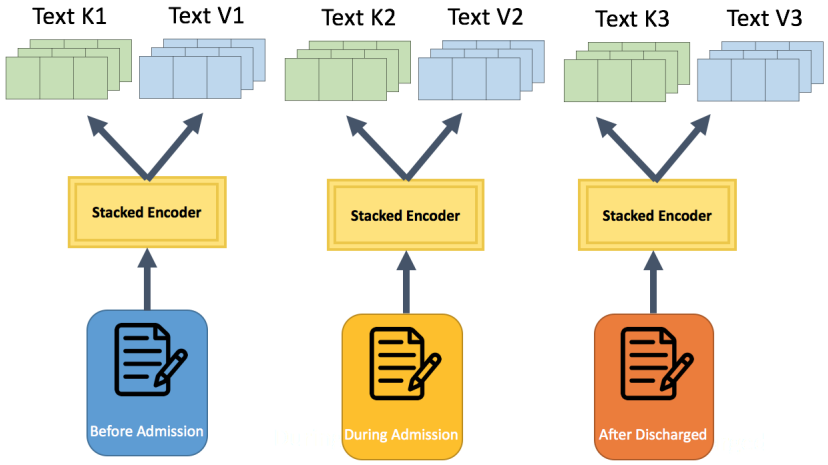

Figure 3.5: Unstructured Representation Module

attention to.

In our model, we implemented an Encoder to process each category of data. Here we followed the common practice to allow a couple of Encoders stacked on top of each other to enhance prediction performance. Thus, in total, there are 3 stacks of Encoders, with each stack handles one type of discharge note. In the end, the stacked Encoder will give a new representation for each word in the document, or in other words, it will generate a matrix $H \in R^{d_{out} \times N}$, where d_{out} is the output dimension of Encoder, and N is the number of words in the document. Each row corresponds to a word representation. For preparation of the following steps, we further project each new representation to a key vector and value vector. Thus, in matrix form, we multiply H^T by $A, B \in R^{d_{out} \times d_{keyval}}$, which yields $TextK, TextV \in R^{N \times d_{keyval}}$. A, B are the transformation matrices that are learned during training time. The entire module is shown in figure 3.5.

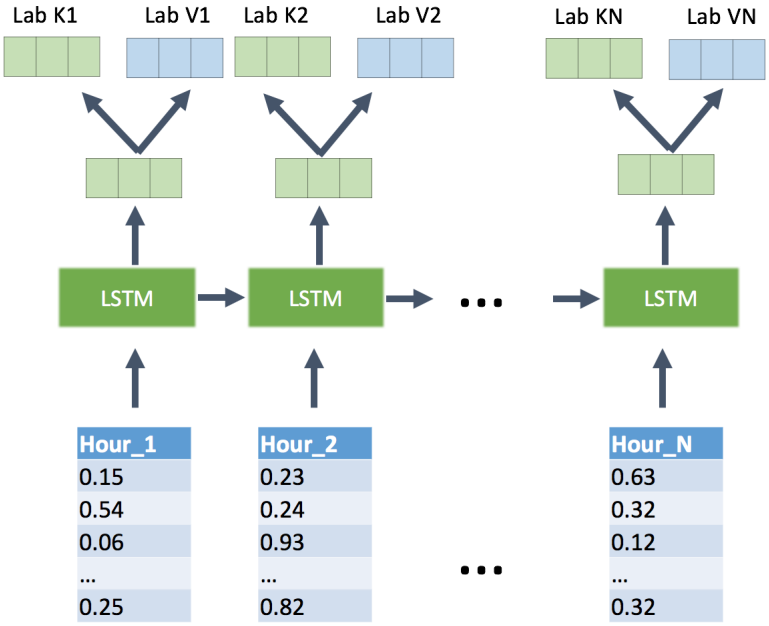

Figure 3.6: Structured Representation Module

Structured representation module

Because of the temporal structure of the lab result and vital sign, we simply concatenate each hour's result and pass each to an LSTM cell orderly. Again, the output for each cell in LSTM will be considered as a new representation for the corresponding hour's input. In the exact same way as we did in the unstructured representation module, we project each new representation into a key and a value vector. Thus, the output of the module for each structured sample will be $LabK, LabV \in R^{24 \times d_{keyval}}$, since we by default, we selected only the first 24 hours lab/vital data for each admission. The module is displayed in figure 3.6.

Attention Linking

In this step, we calculate the attention score between each word in the document and each hour of structured data. The way we calculate the attention score is by using cosine similarity between two vectors, which, in our case, is the dot product between a $labK$ vector (i.e., A row in the $LabK$ matrix), and a $textK$ vector (i.e., A row in the $TextK$ matrix). To do this in matrix form, we multiply $LabK$ by the transpose of $TextK$ matrix, which yields the attention score matrix $A \in R^{24 \times N}$. Lastly, we normalize the matrix by applying Softmax function across rows.

The next step will be obtaining a new and more concise representation of the text by the calculated attention score. To do this, we simply times A by $TextV$, which results in a new matrix $TextAttn \in R^{24 \times d_{keyval}}$. A simplified demonstration of the algorithm above is shown in figure 3.7.

Basically, for each of the 24 hours, we will have three new representation of the text, each represents one of the three text categories. For each of the 24 timestamps, we will concatenate the three text representations along with the lab representation (a row in the $LabV$ matrix), and pass them to 24 LSTM cells accordingly. In this way, we successfully combine structured and unstructured data. This process is represented in figure 3.8. We will move on with the hidden output of the last LSTM cell, H_24.

Lastly, as described in figure 3.9 we concatenate H_24 with the demographic data, and pass the vector through a feed-forward neural network. After normalization, we should be able to obtain a predicted readmission probability.

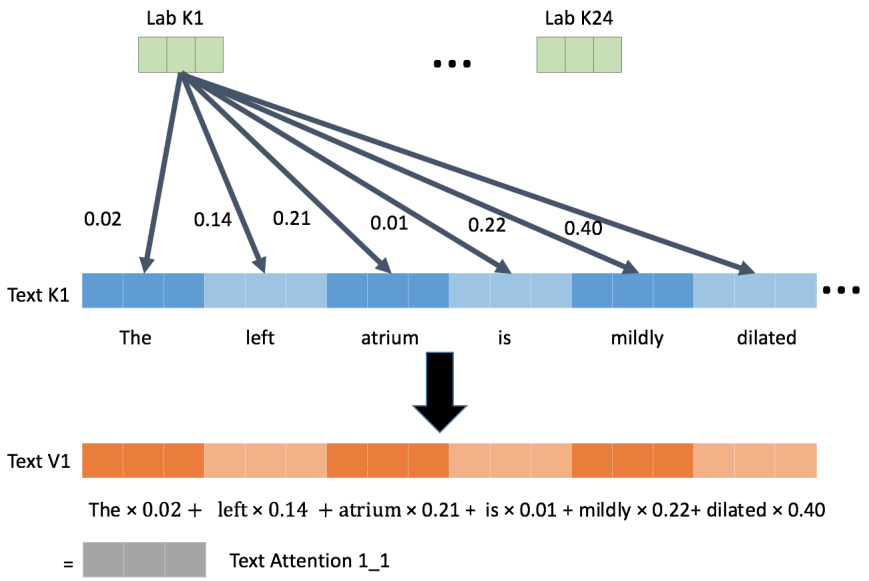

Figure 3.7: Attention Linking Example. In the example, the attention score is calculated between the sentence "The left atrium is mildly dilated" and the structured data for the first hour of admission. The vector "Text Attention 1_1" is the final representation of the text.

Figure 3.8: Combine new text representation with structured representation, and pass them to LSTM

3.4.3 Training Specifications

Same as in 3.3.3, we are using 5-fold cross validation for training-testing split, as well as BCE Loss for loss function of the model.

The word embeddings are entirely learned and updated during the train-

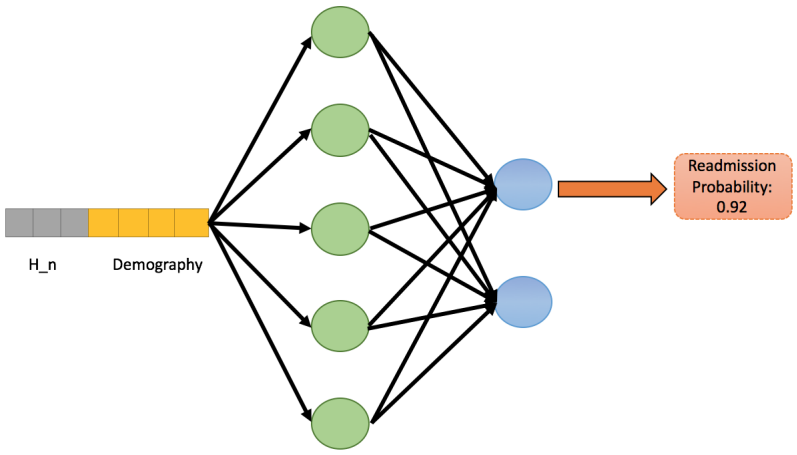

Figure 3.9: Append demographic data, pass to feed-forward network

ing process via back-propagation of the loss, and this is implemented by using PyTorch Embedding Layer. For all LSTM cells being used in this study, we set the dimension of both hidden state and cell state to be 64. Regarding the hyperparameters of Transformer Encoder, the dimension for each feed-forward layer is set to be 1024, while dropout rate is set to be 0.2. Lastly, for attention linking step, all key vectors and value vectors have the size of 32. We chose to have 2 Encoders stacked together for processing each category of documents.

Again, Adam optimizer with weight decay is used for optimizing learning rate, and GPU computing is used for model training.

Chapter 4

Results and Discussion

In this chapter, we will first introduce the metrics we used for the evaluation of the models, as well as the baseline methods we would like to compare with. Then the performance result of the baselines and the two approaches will be displayed. A short discussion of the result will be presented in the end.

4.1 Evaluation Metrics

For evaluation, the following five metrics will be used to judge the performance of the models:

- **Accuracy**: The percentage of correct predictions: $(TruePositives + TrueNegatives)/TotalExamples$
- **Precision**: Ratio of correctly predicted positive observation to the total predicted positive observations: $TruePositives/(TruePositives + FalsePositives)$
- **Recall**: Ratio of correctly predicted positive observations to all positive observations: $TruePositives/(TruePositives + FalseNegatives)$

- **F1 Score**: The weighted average of precision and recall: $2 \cdot Precision \cdot Recall / (Precision + Recall)$
- **AUC-ROC**: AUC-ROC is generally a good measurement for a model's capability of distinguishing between different classes. The number should be between 0 and 1. The higher the AUC, the better the model can distinguish between classes.

Overall, from a Computer Science perspective, F1-Score and AUC-ROC could be most indicative of the general classification performance of a model and are not biased by imbalanced class distributions. On the other hand, for hospitals who do not want to miss any potential positive sample (i.e. A patient who is supposed to be readmmited), recall could be the measurement that they puts more weight on, since recall represents the ability of the model to capture actual positives.

4.2 Baseline Methods

To compare with the 2 approaches, we propose the following baseline methods:

1. **Text Bag-of-words representation**: The clinical notes are represented by word occurrences, and then passed into a logistic regression model for readmission prediction.
2. **Original ICD codes**: The original ICD codes from the MIMIC-III database is encoded by one-hot vector, and then passed into a logistic regression model for readmission prediction.

3. **Generated ICD codes**: The predicted ICD codes by DR-CAML [7] model is used for prediction before passed into logistic regression.

4. **Merged ICD codes**: Both the original and generated ICD codes are combined together, and pass into logistic regression to predict readmission.

5. **Lab/Vital**: First 24 hours lab vital sign data is fed into a LSTM, followed by a fully-connected layer for prediction.

4.3 Model Performance

Table 4.1 displays the performance of our two approaches, as well as the five baselines. In each of the five metrics columns, the number in bold stands for the best result we obtained for the corresponding metric.

	Method	**Acc**	**Prec**	**Rec**	**F1**	**AUC**
Baseline	Text BOW	0.787	0.538	0.580	0.558	0.703
	Original ICD	0.795	0.421	0.637	0.507	0.670
	Generated ICD	0.793	0.424	0.627	0.506	0.670
	Merged ICD	0.793	0.421	0.637	0.507	0.670
	Lab/Vital	0.732	0.475	0.563	0.515	0.535
Approach 1	Implicit Text	0.828	0.514	0.717	0.598	0.723
	Explicit Text	0.827	0.496	**0.724**	0.589	0.717
Approach 2	Attention Linking	**0.867**	**0.773**	0.723	**0.747**	**0.836**

Table 4.1: Model Performance

In short, it can be easily noticed that the Attention Linking network outperforms the other methods in all five metrics but recall. Approach 1

comes next (Definition of "Implicit Text" and "Explicit Text" refers to Figure 3.2) as they both exceed the baselines in terms of accuracy, recall, F1 score, and AUC-ROC. For baseline methods, they all achieve decent accuracy, but Text Bag-of-words representation tops the others regarding precision, F1 score, and AUC-ROC.

4.4 Interpretation of Attention

In the attention-based knowledge linking network, the attention score between each hour's structured data and each word in a document indeed provides some interesting points to investigate as illustrated below in Figure 4.1.

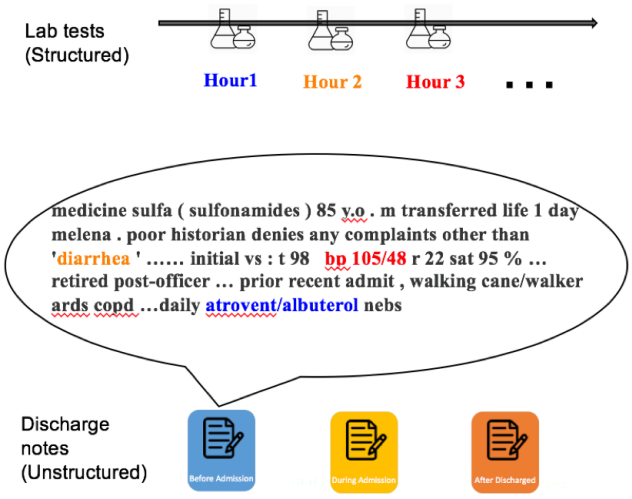

Figure 4.1: Attention Score Interpretation Example

In this example, the attention score between the first three hours' structured data and each word in the 'Before Admission' subtext is examined, and

the word that has highest attention score for each hour's structured data is colored correspondingly. Here, we can see that the first hour's lab tests/vital signs has the tightest relation to the medicine 'atrovent/albuterol', while hour 2's structured input corresponds the most to the symptom 'diarrhea'. The next hour's structured data seems to be mostly related to another aspect, which is the blood pressure abbreviated as 'bp' in the text. This interpretation suggests that the focus between the structured and unstructured data varies as the time progresses. However, this result needs to be verified in the future with the help of clinical experts.

4.5 Discussion

According to the model performance, the two approaches we developed for combining structured and unstructured data indeed demonstrate their effectiveness over the five baseline methods. This suggests that the knowledge contained in both structured and unstructured data are indispensable for predicting hospital readmission.

Also, it is under our expectation that Attention Linking performs the best. The reason could be multifold: First of all, the network architecture is complex enough to process and capture important features in both structured and unstructured data. For example, the encoder network has been proved to be successful in generating better word embeddings for multiple NLP tasks [26]. Secondly, by using attention mechanism between structured and unstructured representations, we enforce the model to focus more on information that are correlated with each other. In this way, the model fuse

the two types of data more "carefully", instead of simply concatenate two vectors together.

For approach 1, both implicit text and explicit text representation turn out to be comparable to each other in terms of the five metrics. Implicit text representation seems to be slightly better in all metrics but recall. Intuitively this makes sense, since matrix representation may contain more subtle information than pure probability representation.

Additionally, the attention-based knowledge linking network provides some valuable insights of how the focus or relation change between each hour's structured data and each word in the unstructured text.

Method	**Training Time (per epoch)**	**Prediction Time (per sample)**
Approach 1	$\sim 3min$	<1 sec
Approach 2	$\sim 2min$	<1 sec

Table 4.2: Model Running Time

Last but not least, regarding the running time (on GPU) of the two approaches, Approach 1 (referring to both implicit and explicit representations) takes about three minutes for training an epoch, while generally, the model takes about 30 epochs to converge. Therefore, the total training time is approximately one and a half hours. On the other hand, approach 2 takes a little less than 2 minutes for an epoch, and it usually requires 20 epochs to terminate training. Thus, it will be around 40 minutes in total. The prediction result for both approaches on a single sample can be obtained within a second. Thus, since training will not occur as frequently in practical sce-

narios, we argue that the two approaches should be feasible for real-world usage.

Chapter 5

Conclusion

In this paper, we present two novel approaches effectively combine structured clinical data with unstructured medical notes, to predict 30-day hospital readmission probability. Both of the approaches achieve superior results compared with the five baselines, which use only structured or unstructured data. Comparing the two approaches, Knowledge Linking turns out to be better, as it combines structured and unstructured representation in a more deliberate manner.

In the future, we may consider improving the interpretability of the knowledge-linking network, since right now, there are still plenty of attention scores that do not make a lot of sense. Thus, how to make attention score stable enough for interpretation purposes will be an interesting area to look at. Besides, we may consider generalizing these methods to more clinical prediction tasks such as patient's in-hospital mortality prediction and hospital length-of-stay forecasting. The concept of attention-based knowledge linking method may also be generalized to other fields of study, as long as both structured and unstructured data are presented. Another thing we may consider in the future is to incorporate more types of data in clinical prediction tasks, such as MRI image data, or perhaps patient's speech data, to make the prediction model more comprehensive and accurate.

Bibliography

[1] A. Campbell *et al.*, "Predicting death and readmission after intensive care discharge," *BJA: British Journal of Anaesthesia*, Apr. 2, 2008.

[2] H. Baek *et al.*, "Analysis of length of hospital stay using electronic health records: A statistical and data mining approach," *PLOS One*, Apr. 13, 2018.

[3] B. Shickel *et al.*, "Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE Journal of Biomedical and Health Informatics*, Sep. 2018.

[4] Y. Bengio *et al.*, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Apr. 2014.

[5] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1721–1730.

[6] J. Liu, Z. Zhang, and N. Razavian, "Deep ehr: Chronic disease prediction using medical notes," *Journal of Machine Learning Research (JMLR)*, 2018.

[7] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, ACL Anthology, 2018, 1101–1111.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[10] Z. Xia, E. Secor, L. Chibnik, R. Bove, and S. e. e. Cheng, "Modeling disease severity in multiple sclerosis using electronic health records," *Plus One*, 2013.

[11] S. Wu, S. Sohn, and K. e. e. Ravikumar, "Automated chart review for asthma cohort identification using natural language processing: An exploratory study," *Annals of Allergy, Asthma Immunology*, 2013.

[12] M. Bisbal, E. Jouve, L. Papazian, S. de Bourmont, G. Perrin, B. Eon, and M. Gainnier, "Effectiveness of saps iii to predict hospital mortality for post-cardiac arrest patients," *Resuscitation*, vol. 85, no. 7, pp. 939–944, 2014.

[13] J. Donze, D. Aujesky, D. Williams, and J. L. Schnipper, "Potentially avoidable 30-day hospital readmissions in medical patients: Derivation and validation of a prediction model," *JAMA internal medicine*, vol. 173, no. 8, pp. 632–638, 2013.

[14] A. B. Nassif, I. Shahin, I. Attili, M. Azzehm, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," vol. 60, no. 6, 84–90, 2017.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, MIT Press, 2014, 3104–3112.

[18] M. Sushil, S. Šuster, K. Luyckx, and W. Daelemans, "Patient representation learning and interpretable evaluation using clinical notes," *Journal of biomedical informatics*, vol. 84, pp. 103–113, 2018.

[19] F. Tang, C. Xiao, F. Wang, and J. Zhou, "Predictive modeling in urgent care: A comparative study of machine learning approaches," *JAMIA Open*, vol. 1, no. 1, pp. 87–98, 2018.

[20] H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data.," *CoRR*, vol. abs/1703.07771, 2017.

[21] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets.," *J. Biomed. Informatics*, vol. 83, pp. 112–134, 2018.

[22] J. Mengqi, T. B. Mohammad, C. Aaron, B. Parminder, and C. e. e. Busra,

[23] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160 035, 2016.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*.

[25] O. S. Center, *Ohio supercomputer center*, 1987. [Online]. Available: `http://osc.edu/ark:/19495/f5s1ph73`.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.